

TIPI DI VARIABILI IN C

I tipi di dati scalari in C sono 3:

Tipo intero

Tipo reale

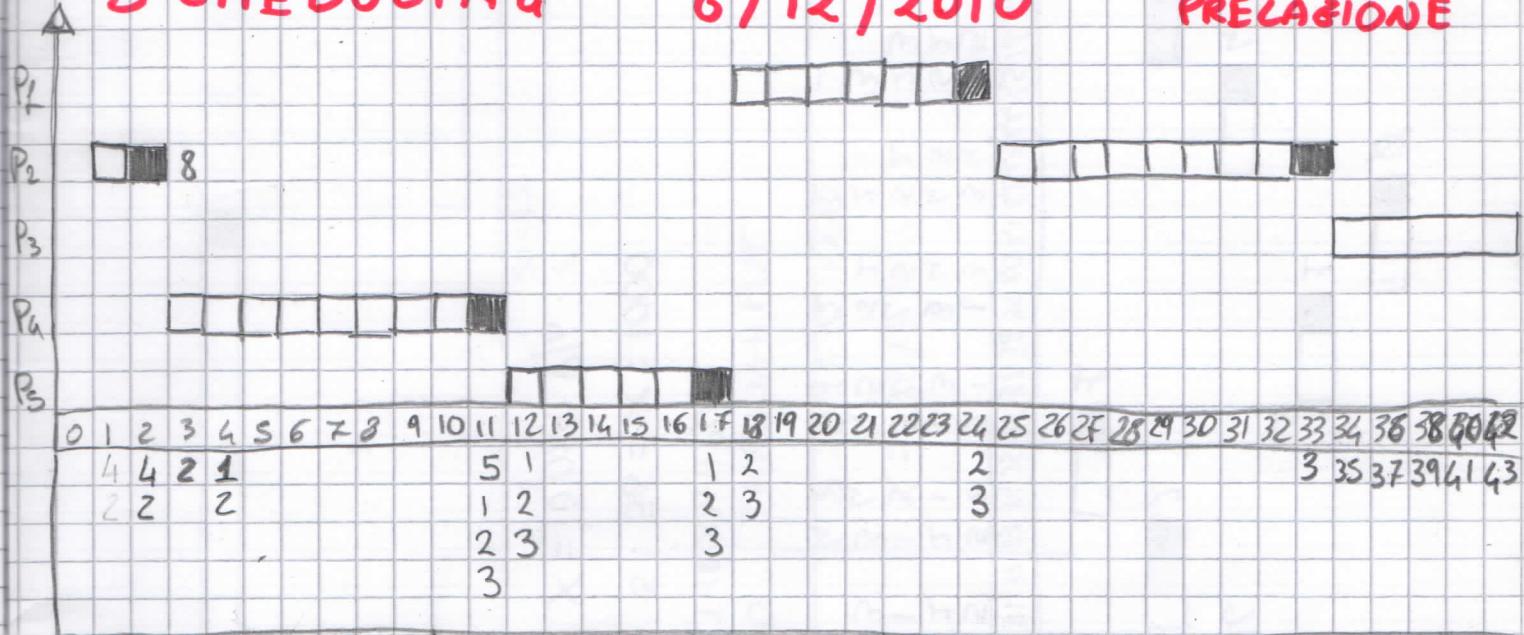
Tipo carattere

Per definire nuovi tipi di dati scalari

SCHEDULING

6/12/2010

PRIORITÀ CON PRELACIONE



ATTESA

$$\begin{aligned}
 P1 &= 18 - 4 = 14 \text{ ms} \\
 P2 &= 25 - 2 = 23 \text{ ms} \\
 P3 &= 34 - 6 = 28 \text{ ms} \\
 P4 &= 3 - 2 = 1 \text{ ms} \\
 PS &= 12 - 7 = 5 \text{ ms} \\
 \text{MEDIO} &= 14,2 \text{ ms}
 \end{aligned}$$

TURNAROUND

$$\begin{aligned}
 PT &= 14 + 6 = 20 \text{ ms} \\
 P2 &= 23 + 9 = 32 \text{ ms} \\
 P3 &= 28 + 10 = 38 \text{ ms} \\
 P4 &= 1 + 8 = 9 \text{ ms} \\
 PS &= 5 + 5 = 10 \text{ ms} \\
 \text{MEDIO} &= 21,8 \text{ ms}
 \end{aligned}$$

TA NORMALIZZATO

$$\begin{aligned}
 P1 &= 20 / 6 = 3,33 \\
 P2 &= 32 / 9 = 3,55 \\
 P3 &= 38 / 10 = 3,8 \\
 P4 &= 9 / 8 = 1,13 \\
 PS &= 10 / 5 = 2 \\
 \text{MEDIO} &= 2,76
 \end{aligned}$$

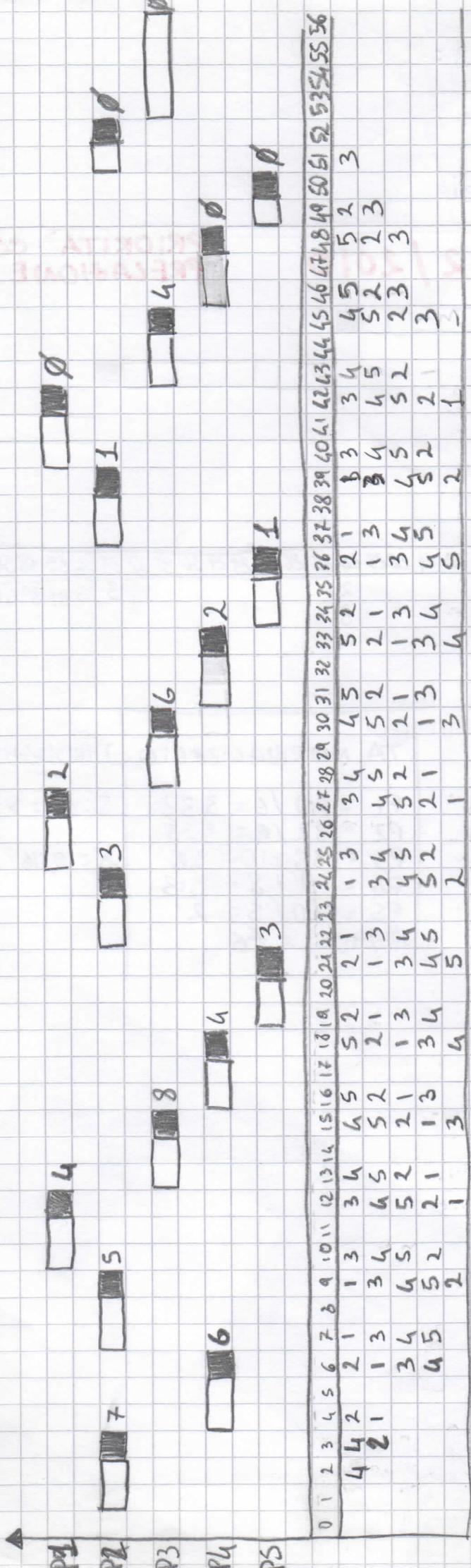
$$\begin{aligned}
 5 : 13 &= x : 100 \\
 x &= 0,116
 \end{aligned}$$

PROCESSO	TEMPO ARRIVO	EPO BURST	PRIORITA'
P1	4 ms	6 ms	3
P2	1 ms	9 ms	2
P3	6 ms	10 ms	1
P4	2 ms	8 ms	5
P5	7 ms	5 ms	4

SCHEDULING 6/12/2010

ROUND ROBIN

(è un algoritmo con prelievo)



$$\begin{aligned}
 P1 &= 1 - 3 = 6 = 32 \text{ ms} & P1 &= 32 + 6 = 38 \text{ ms} & P1 &= 38 / 6 = 6,3 \\
 P2 &= 5 - 1 = 4 = 42 \text{ ms} & P2 &= 42 + 9 = 51 \text{ ms} & P2 &= 51 / 4 = 5,4 \\
 P3 &= 56 - 5 - 10 = 11 = 51 \text{ ms} & P3 &= 51 + 10 = 61 \text{ ms} & P3 &= 61 / 11 = 5,1 \\
 P4 &= 47 - 1 - 8 = 38 = 46 \text{ ms} & P4 &= 38 + 9 = 46 \text{ ms} & P4 &= 46 / 8 = 5,8 \\
 P5 &= 6 - 5 = 38 = 43 \text{ ms} & P5 &= 38 + 5 = 43 \text{ ms} & P5 &= 43 / 5 = 8,6 \\
 \text{MEDIO} &= 38,2 & \text{MEDIO} &= 45,8 \text{ ms} & \text{MEDIO} &= 6,3
 \end{aligned}$$

$$\begin{aligned}
 5 : 56 = X : 1000 \\
 X = 0,089 \text{ P/D}
 \end{aligned}$$

THROUGHPUT

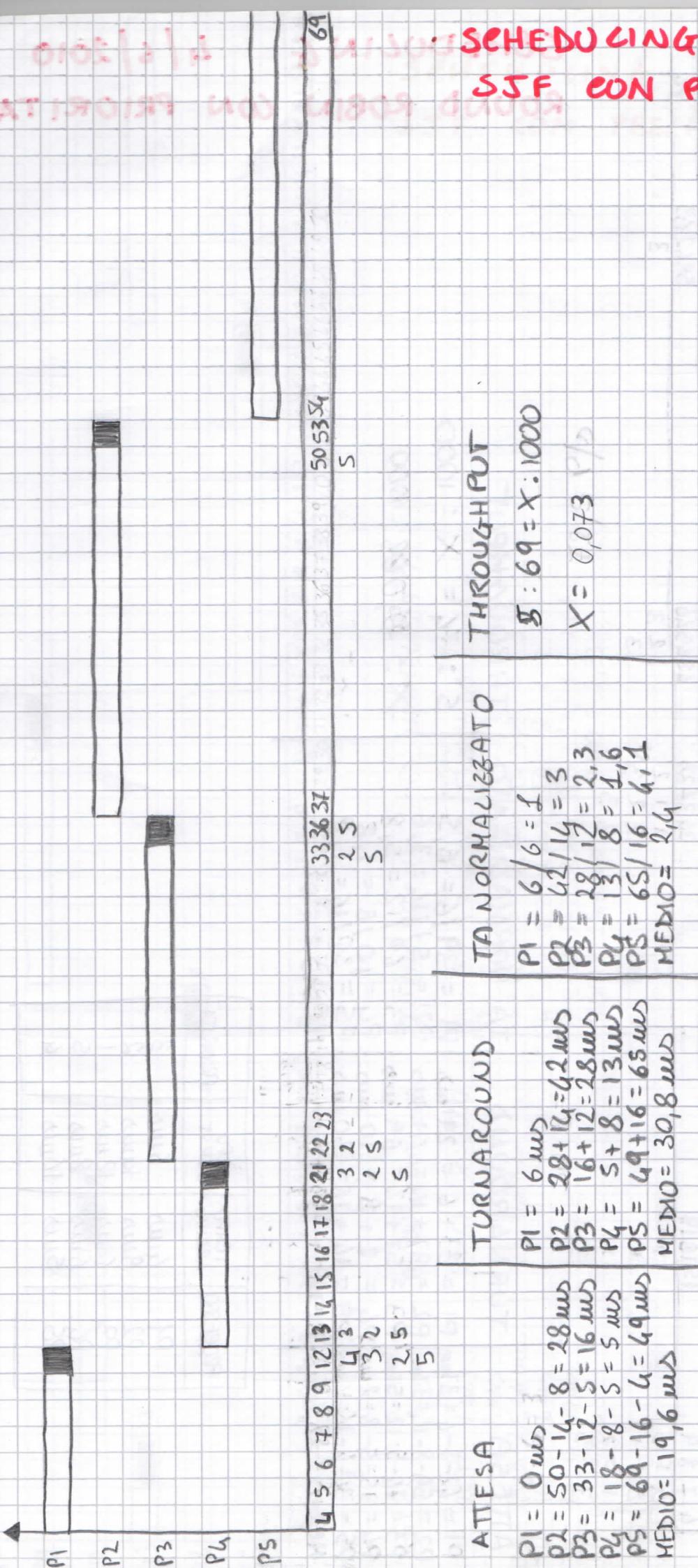
$$\begin{aligned}
 P1 &= 38 / 6 = 6,3 \\
 P2 &= 51 / 4 = 5,4 \\
 P3 &= 61 / 11 = 5,1 \\
 P4 &= 46 / 8 = 5,8 \\
 P5 &= 43 / 5 = 8,6 \\
 \text{MEDIO} &= 6,3
 \end{aligned}$$

PROCESSO	TEMPO ARRIVO	CPU BURST	PERIODICITA'
P1	4 ms	6 ms	3
P2	7 ms	9 ms	2
P3	6 ms	10 ms	1
P4	2 ms	8 ms	5
P5	7 ms	5 ms	4

SCHEDULING

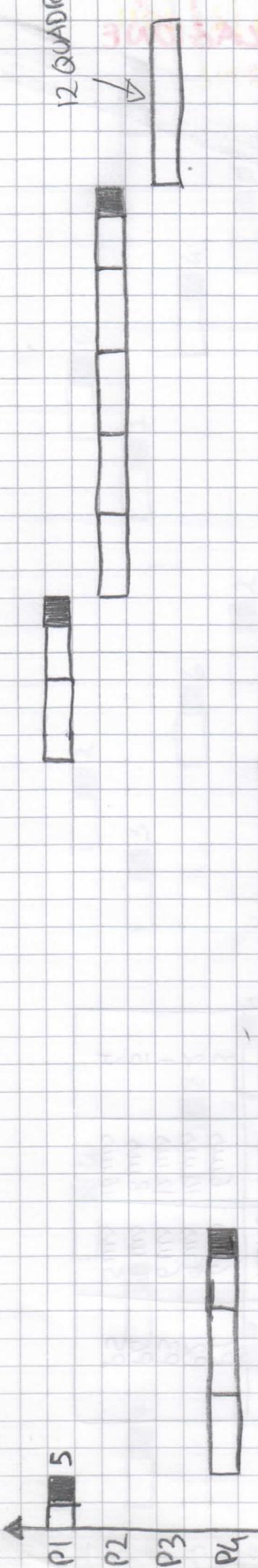
4/6/2010

SJF CON PRELACIONE



	PROCESSO	TEMPO ARRIVO	CPO Bj REST	PRIORITA
	P1	6 min	6 min	3
	P2	8 min	16 min	2
	P3	6 min	12 min	1
	P4	6 min	8 min	5
	P5	5 min	16 min	4

12 QUADRATI



	P1	P2	P3	P4	P5	Turnaround	Waiting	Throughput
P1	33 ms	37 ms	57 ms	15 ms	31 ms	39 ms	33 ms	5 : $T_4 = X : 1000$
P2	37 ms	57 ms	69 ms	2 ms	10 ms	42 ms	37 ms	$X = 0,068$
P3	57 ms	69 ms	81 ms	1 ms	30 ms	79 ms	57 ms	
P4	15 ms	2 ms	10 ms	1 ms	30 ms	16 ms	15 ms	
P5	31 ms	10 ms	30 ms	1 ms	16 ms	46 ms	31 ms	
	4	7	8	9	5	15	18	19
	5	4	3	2	1	5	4	3

SCHEDULING 4/6/2010

ROUND ROBIN CON PRIORITA'

TA NORMAIZZATO

TA NORMAIZZATO

TURNAROUND

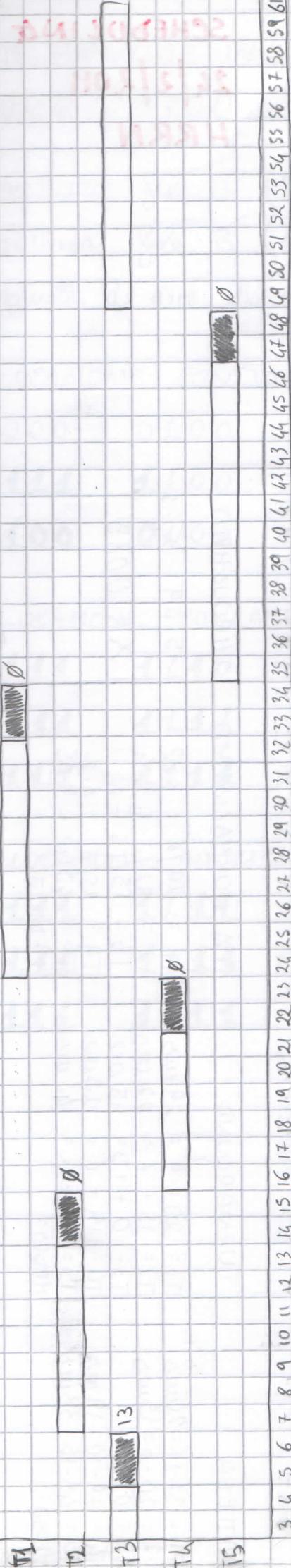
ATTESA

$$\begin{aligned}
 P1 &= 12 - 3 - 6 = 33 \text{ ms} & P1 &= 33 + 6 = 39 \text{ ms} & P1 &= 39 / 6 = 6,5 \\
 P2 &= 59 - 8 - 14 = 37 \text{ ms} & P2 &= 37 + 4 = 51 \text{ ms} & P2 &= 51 / 14 = 3,6 \\
 P3 &= 74 - 5 - 12 = 57 \text{ ms} & P3 &= 57 + 12 = 69 \text{ ms} & P3 &= 69 / 12 = 5,8 \\
 P4 &= 15 - 5 - 8 = 2 \text{ ms} & P4 &= 2 + 8 = 10 \text{ ms} & P4 &= 10 / 8 = 1,3 \\
 P5 &= 34 - 4 - 16 = 16 \text{ ms} & P5 &= 16 + 16 = 32 \text{ ms} & P5 &= 32 / 16 = 2,0 \\
 \text{MEDIO} &= 28,6 \text{ ms} & \text{MEDIO} &= 39,8 \text{ ms} & \text{MEDIO} &= 3,8
 \end{aligned}$$

PROCESSO	TEMPO ARRIVO	EPO BURST	PRIORITA'
P1	4 ms	6 ms	3
P2	9 ms	14 ms	2
P3	6 ms	12 ms	1
P4	6 ms	8 ms	5
P5	5 ms	16 ms	4

SCHEDULING 24/2/2011

SJF CON PRELAZIONE



3 6 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 61

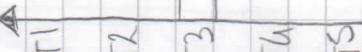
ATTESA	THROUGHPUT	TURNAROUND	TA NORMALIZZATO
$T_1 = 24 - 17 = 7 \text{ ms}$	$S : 61 = X : 1000$	$T_1 = 7 + 9 = 16 \text{ ms}$	$T_1 = 16 / 9 = 1,78$
$T_2 = 7 - 5 = 2 \text{ ms}$		$T_2 = 2 + 7 = 9 \text{ ms}$	$T_2 = 9 / 7 = 1,29$
$T_3 = 19 - 13 - 2 = 4 \text{ ms}$	$X = 0,082$	$T_3 = 44 + 15 = 59 \text{ ms}$	$T_3 = 59 / 15 = 3,93$
$T_4 = 16 - 10 = 6 \text{ ms}$		$T_4 = 6 + 6 = 12 \text{ ms}$	$T_4 = 12 / 6 = 2$
$T_5 = 35 - 15 = 20 \text{ ms}$		$T_5 = 20 + 12 = 32 \text{ ms}$	$T_5 = 32 / 12 = 2,67$
$MEDIO = 15,8 \text{ ms}$		$MEDIO = 25,6 \text{ ms}$	$MEDIO = 2,33$

$$\begin{aligned}
 T_1 &= 24 - 17 = 7 \text{ ms} \\
 T_2 &= 7 - 5 = 2 \text{ ms} \\
 T_3 &= 19 - 13 - 2 = 4 \text{ ms} \\
 T_4 &= 16 - 10 = 6 \text{ ms} \\
 T_5 &= 35 - 15 = 20 \text{ ms} \\
 \text{MEDIO} &= 15,8 \text{ ms}
 \end{aligned}$$

$$\begin{aligned}
 S : 61 &= X : 1000 \\
 X &= 0,082
 \end{aligned}$$

$$\begin{aligned}
 T_1 &= 7 + 9 = 16 \text{ ms} \\
 T_2 &= 2 + 7 = 9 \text{ ms} \\
 T_3 &= 44 + 15 = 59 \text{ ms} \\
 T_4 &= 6 + 6 = 12 \text{ ms} \\
 T_5 &= 20 + 12 = 32 \text{ ms} \\
 \text{MEDIO} &= 25,6 \text{ ms}
 \end{aligned}$$

$$\begin{aligned}
 T_1 &= 7 + 9 = 16 \text{ ms} \\
 T_2 &= 2 + 7 = 9 \text{ ms} \\
 T_3 &= 44 + 15 = 59 \text{ ms} \\
 T_4 &= 6 + 6 = 12 \text{ ms} \\
 T_5 &= 20 + 12 = 32 \text{ ms} \\
 \text{MEDIO} &= 2,33
 \end{aligned}$$



3 4 5 6 7 8 9 10 " 12 13 - - T 18 19 20

26 27 28 29
34 35 36 37

45464748

$$RR\text{TR}_1 = \frac{3+9}{9} = 1,33 \quad RR\text{TR}_2 = \frac{15+7}{7} = 3,14 \quad RR\text{TS} = \frac{3+12}{12} = 1,42 \quad ISTANTE 19$$

$$RRTu = \frac{10+6}{6} = 2,67 \quad RRTS = \frac{3+12}{12} = 1,42$$

INSTANT 19

$$RRT1 = \frac{12+9}{9} = 2_1$$

$$RRTS = \frac{14+12}{12} = 2,17$$

INSTANCE 28

$$RRT_4 = \frac{20+9}{9} = 3,22 \quad RRTS = \frac{22+12}{12} = 2,82$$

15 ANTE 36

ATESEA

TURNAROUND

DATA NORMALISATION THROUGHPUT

50

$$\begin{aligned}
 T1 &= 37 - 17 = 20 \text{ ms} \\
 T2 &= 20 - 5 = 15 \text{ ms} \\
 T3 &= 0 \text{ ms} \\
 T4 &= 29 - 10 = 19 \text{ ms} \\
 T5 &= 68 - 15 = 33 \text{ ms} \\
 \text{MFDIO} &=
 \end{aligned}$$

$$\begin{aligned}
 11 &= 20 + 9 = 29 \text{ ms} \\
 12 &= 15 + 7 = 22 \text{ ms} \\
 3 &= 0 + 15 = 15 \text{ ms} \\
 4 &= 19 + 6 = 25 \text{ ms} \\
 5 &= 33 + 12 = 45 \text{ ms}
 \end{aligned}$$

(F1D10 =

$$\begin{aligned}
 \text{M} &= 20/9 = 3,22 \\
 \text{T} &= 22/7 = 3,14 \\
 \text{B} &= 15/15 = 1 \\
 \text{Y} &= 25/6 = 4,17 \\
 \text{S} &= 45/12 = 3,75 \\
 \text{MSBIO} &= 10
 \end{aligned}$$

$$X = 0.085$$

MEDULING

SCHEDULING
24/2/2011
HRRN

CONFIGURAZIONE INIZIALE

BITMAP 6/3/09

1000 0100 1010 0010 0000 0010

	BLOCCHI CANELLATI
	BLOCCHI A
	BLOCCHI B
	INDICE A
	INDICE B

a) scrittura file A (5 blocchi)

b) scrittura file B (4 blocchi)

c) aggiunta di due blocchi ad A, cancellazione di tre blocchi a B

ALLOCAZIONE CONIGUA (gli 1 devono essere successivi)

a) 1000 0100 1010 0011 1111 0010

b) 1 111 1100 1010 0011 1111 0010

c) 1 100 0100 1010 0011 1111 1110

ALLOCAZIONE CONCATENATA (gli 1 possono non essere successivi)

a) 1 111 1110 1010 0010 0000 0010

b) 1 111 1111 1111 1010 0000 0010

c) 1 111 1111 1010 0111 0000 00010

ALLOCAZIONE INDIRETTA (come la concatenata solo che c'è bisogno dell'inoltre)

a) 1 111 1110 1010 0010 0000 0010

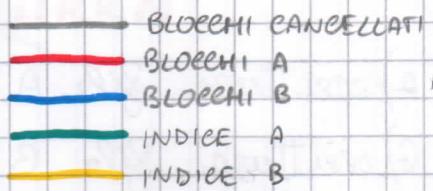
b) 1 111 1111 1111 1111 0000 0010

c) 1 111 1111 1111 0010 11 0000 0010

CONFIGURAZIONE INIZIALE

BITMAP 30/1/09

1000 1000 1000 0010



- scrittura file A (2 blocchi)
- scrittura file B (3 blocchi)
- aggiunta di due blocchi ad A, cancellazione di B

ALLOCAZIONE CONTIGUA

a) 110 1000 1000 0010

b) 110 111 1000 0010

c) NON ALLOCABILE

ALLOCAZIONE CONCATENATA

a) 110 1000 1000 0010

b) 111 110 1000 0010

c) 110 1001 100 0010

ALLOCAZIONE INDIRETTA

a) 111 1000 1000 0010

b) 111 111 100 10010

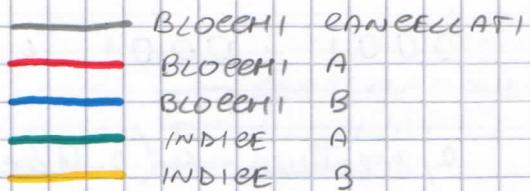
c) 111 1000 1011 00010

CONFIGURAZIONE INIZIALE

BITMAP 24-2-2011

1000 0011 0100 0000 0110 1010

- a) scrittura A (6 bloechi)
- b) scrittura B (4 bloechi)
- c) aggiunta di due bloechi ad A
con cancellazione di B



ALLOCAZIONE CONTIGUA

- a) 1000 0011 01~~11~~ ~~1111~~ 0110 1010
- b) 1~~111~~ ~~1~~011 01~~11~~ ~~1111~~ 0110 1010
- c) NON ALLOCABILE

ALLOCAZIONE CONCATENATA

- a) 1~~111~~ ~~11~~11 1100 0000 0110 1010
- b) 1~~111~~ ~~11~~11 ~~1111~~ ~~1100~~ 0110 1010
- c) 1~~111~~ ~~11~~11 ~~1100~~ ~~0011~~ 0110 1010

ALLOCAZIONE INDIRETTA

- a) 1~~111~~ ~~11~~11 ~~110~~ 0000 0110 1010
- b) 1~~111~~ ~~11~~11 ~~111~~ ~~1111~~ 0110 1010
- c) 1~~111~~ ~~11~~11 ~~1110~~ ~~0000~~ ~~1111~~ 1010

CONFIGURAZIONE INIZIALE

0001	0001	1000	0010
		<u> </u>	
		<u> </u>	

- CANCELLAZIONE
- BLOCCHI A
- BLOCCHI B
- INDICE A
- INDICE B

- a) scrittura A (2 Blocchi)
- b) scrittura B (3 Blocchi)
- c) aggiunta due blocchi ad A, cancellazione B

ALLOC. CONTIGUA

11 01	0001	1000	0010
		<u> </u>	
		<u> </u>	

11 01	111 1	1000	0010
		<u> </u>	
		<u> </u>	

- e) NON ALLOCABILE

ALLOC. CONSECUTIVA

11 01	0001	1000	0010
		<u> </u>	
		<u> </u>	

111 1	11 01	1000	0010
		<u> </u>	
		<u> </u>	

110 1	001 1	1 1 00	0010
		<u> </u>	
		<u> </u>	

ALLOC. INDICIZZATA

111 1	0001	1000	0010
		<u> </u>	
		<u> </u>	

111 1	111 1	1 1 00	0010
		<u> </u>	
		<u> </u>	

111 1	0001	10 1 1	0010
		<u> </u>	
		<u> </u>	

30/08/08 SANTIS

BITHAP 31/3/08

CONFIGURAZIONE INIZIALE

0 0 0 0 1 1 0 0 1 0 0 1 1 0 1 0

- CANCELLAZIONE
- BLOCCO A
- BLOCCO B
- INDIRETTO A
- INDIRETTO B

- a) scrittura A (4 blocchi)
- b) scrittura B (3 blocchi)
- c) aggiunta 2 blocchi A, cancellazione B

ALLOCAZIONE CONIGUA

- a) **1 1 1 1** 1 1 0 0 1 0 0 1 1 0 1 0
- b) NON ALLOCABILE
- c) NON ALLOCABILE

ALLOCAZIONE CONCATENATA

- a) **1 1 1 1** 1 1 0 0 1 0 0 1 1 0 1 0
- b) **1 1 1 1** 1 1 **1 1** 1 **1 0 1** 1 0 1 0
- c) **1 1 1 1** 1 1 0 0 1 0 **1 1** 1 **1** 1 0

ALLOCAZIONE INBLIBBATA

- a) **1 1 1 1** 1 1 **1 0** 1 0 0 1 1 0 1 0
- b) **1 1 1 1** 1 1 **1 1** 1 **1 1** 1 1 **1** 1 0
- c) **1 1 1 1** 1 1 **1 1** 1 **1 1** 1 1 **1** 1 0

BOLENEE BANTIE

BITMAP 3/3/08

CONFIG INIZIALE

1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 0

- a) scritt A (4 Blocchi)
- b) scritt B (3 Blocchi)
- c) aggiunto 2 Blocchi A, cancellazione B

CANCELLAZIONE
 BLOCCHI A
 BLOCCHI B
 INDICE A
 INDICE B

ALLOCA. CONTIGUA

- a) 1 0 0 0 1 1 ~~1 1~~ ~~1 1~~ ~~1 1~~ 0 0 1 0 1 0
- b) 1 ~~1 1 1~~ 1 1 ~~1 1~~ ~~1 1~~ ~~1 1~~ 0 0 1 0 1 0
- c) 1 0 0 0 1 1 ~~1 1~~ ~~1 1~~ ~~1 1 1 1~~ 1 0 1 0

ALLOCA. CONCATENATA

- a) 1 ~~1 1 1~~ 1 1 ~~1 0~~ 0 0 0 0 1 0 1 0
- b) 1 ~~1 1 1~~ 1 1 ~~1 1~~ ~~1 1~~ 0 0 1 0 1 0
- c) 1 ~~1 1 1~~ 1 1 ~~1 0~~ 0 0 ~~1 1~~ 1 0 1 0

ALLOCA. INDIREZZATA

- a) 1 ~~1 1 1~~ 1 1 ~~1 1~~ 0 0 0 0 1 0 1 0
- b) 1 ~~1 1 1~~ 1 1 ~~1 1~~ ~~1 1 1 1~~ 1 0 1 0
- c) 1 ~~1 1 1~~ 1 1 ~~1 1~~ 0 0 0 0 ~~1 1 1~~

1024

24

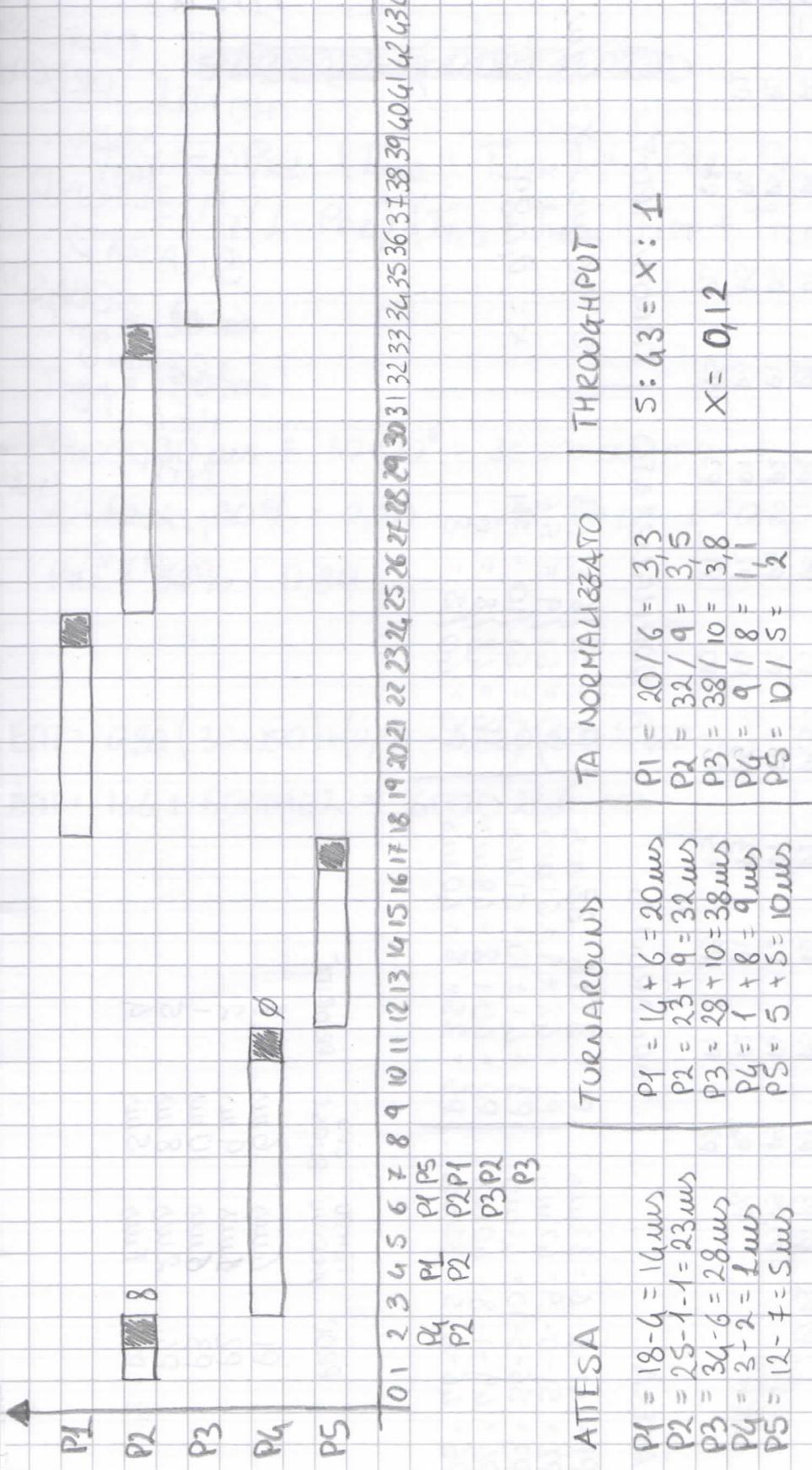
5

4

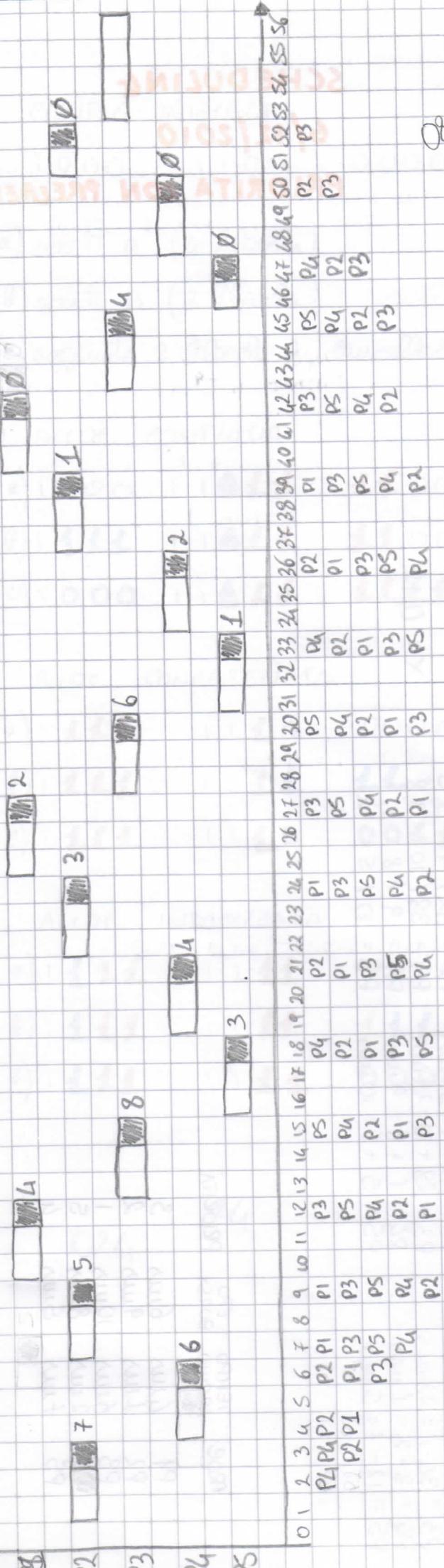
SCHEDULING

6/12/2010

PRIORITA CON PRELAZIONE



PROC	TERPO ARRIVO	EPO QUEST	PERIODA
P1	4 min	6 min	3
P2	1 min	9 min	2
P3	6 min	10 min	1
P4	2 min	8 min	5
P5	7 min	5 min	4



ATTESA

$$\begin{aligned}
 P1 &= 41 - 3 = 38 \\
 P2 &= 51 - 0 = 51 \\
 P3 &= 56 - 5 = 51 \\
 P4 &= 49 - 1 = 48 \\
 P5 &= 46 - 6 = 40
 \end{aligned}$$

$$\begin{aligned}
 p_1 &= 32 + 6 = 38 \text{ ms} \\
 p_2 &= 42 + 9 = 51 \text{ ms} \\
 p_3 &= 41 + 16 = 51 \text{ ms} \\
 p_4 &= 40 + 8 = 48 \text{ ms} \\
 p_5 &= 35 + 5 = 40 \text{ ms}
 \end{aligned}$$

$$\begin{aligned}
 p_1 &= 53/6 = 8 \frac{5}{6} \\
 p_2 &= 51/9 = 5 \frac{6}{9} \\
 p_3 &= 51/10 = 5 \frac{1}{10} \\
 p_4 &= 48/8 = 6 \frac{0}{8} \\
 p_5 &= 40/5 = 8 \frac{0}{5}
 \end{aligned}$$

$$S: 26 \text{ m/s} = x : 1 \text{ m/s}$$

THROUGHPUT

TA NORWAY 200

TURNAROUND

	TEMP	TEMP ACRIVO	CPO BURST	PROLITA
P1	4 min	6 min	9 min	3
P2	5 min	7 min	10 min	2
P3	6 min	8 min	12 min	1
P4	2 min	4 min	5 min	5
P5	3 min	5 min	7 min	4

SCHEDULING

6/12/2010

ROUND ROBIN

SAN GENNARO
SANT'ANASTASIA
MONTE PROCIDA
POZZUOLI
GIUGLIANO
QUARTO
SAN SEBASTIA
ARBANDA
HELITO
NAPOLI
NAPOLI (TO)
VOMERO

In un sistema che usa paginazione, l'accesso al TLB richiede 30 ms, mentre l'accesso alla memoria sia richiede 150 ms.

EAT e TLB

31/01/2011

- a) Quando si verifica un page fault, si perdono 30 ms per correggere la pagina che si sta cercando in memoria. Se il page fault rate è il 20%, il TLB rate è l'80%, indicare l'EAT ai dati

~~(EAT + TTLB + THMEM + TPF + 2THMEM)~~

~~(correggere la pagina) costato~~

$$1 \mu s = 10^{-3} s$$

$$1 ms = 10^{-3} s$$

$$T_{EAT} = P_{R2} (T_{TLB} + T_{MEM}) + (P_{R1} - P_{R2}) (T_{TLB} + 2 T_{MEM}) + (1 - P_{R1}) (T_{TLB} + T_{MEM} + T_{PF} + T_{TLB} + 2 T_{MEM})$$

$$T_{TLB} = 30 \text{ ms}$$

$$T_{MEM} = 150 \text{ ms}$$

$$T_{PF} = 30 \mu s = 30 \times 10^6 = 30\,000\,000 \text{ ns}$$

$$1 - P_{R1} = 20\% = 0,20 \Rightarrow P_{R1} = 1 - 0,20 = 0,80$$

$$P_{R2} = 80\% = 0,80$$

$$EAT = 0,80 (30 + 150) + (0,80 - 0,80) (30 + 2 \cdot 150 \text{ ms}) + 0,20 (30 + 150 + 30\,000\,000 + 30 + 30)$$

$$EAT = 164 + 6000102 = 6000246 \text{ ms}$$

b) Si supponga ora che, quando la ricerca nel TLB ha successo, la pagina richiesta si trovi effettivamente in RAM, con il TLB che ha un miss ratio del 20%. Si sa che il 5% degli indirizzi logici generati provoca un page fault, e nel 60% dei casi di page fault la pagina vittima ha il dirty bit a 1. Il tempo medio necessario per trasferire una pagina dalla RAM all'area di swap (o viceversa) è di 50 millisecondi. Inoltre l'EAT ai dati.

$$\begin{aligned} EAT = & P_{R2} (T_{TLB} + T_{MEM}) + (P_{R2} - P_{R1}) (T_{TLB} + 2T_{MEM}) + \\ & + (1 - P_{R2}) \cdot \%_{DIRTY_ZERO} \cdot (T_{TLB} + T_{MEM} + T_{PF} + T_{TLB} + 2T_{MEM}) + \\ & + (1 - P_{R2}) \cdot \%_{DIRTY_UNO} \cdot (T_{TLB} + T_{MEM} + T_{PF} + T_{SWAP} + T_{TLB} + 2T_{MEM}) \end{aligned}$$

$$\begin{aligned} EAT = & 0,80 (30 + 150) + (0,80 - 0,80) (30 + 2 \cdot 150) + \\ & + 0,20 \cdot 0,40 \cdot (30 + 150 + 30 \cdot 000 \cdot 000 + 30 + 2 \cdot 150) + \\ & + 0,20 \cdot 0,60 \cdot (30 + 150 + 30 \cdot 000 \cdot 000 + 50 \cdot 000 \cdot 000 + 30 + 2 \cdot 150) \end{aligned}$$

$$EAT = 144 + 0 + 2400040,8 + 9600061,2 = 12000266 \text{ ms}$$

$$\%_{DIRTY_ZERO} = 40\% = 0,40$$

$$\%_{DIRTY_UNO} = 60\% = 0,60$$

$$T_{SWAP} = 50 \text{ ms} = 50 \cdot 000 \cdot 000 \text{ ns}$$

~~000~~

In un sistema che usa paginazione, l'accesso al TLB richiede 30 ms, l'accesso alla memoria 350 ms.

EAT & TLB

16/03/2011

a) Quando si verifica un page fault, si perdeva 28 ms per caricare la pagina che si sta creando in memoria. Se il page fault rate è del 18%, il TLB hit rate è il 68%, indicare l'EAT.

$$T_{TLB} = 30 \text{ ms}$$

$$\%_{DIRTRBIT_ZERO} = 60\% = 0,6$$

$$T_{MEM} = 350 \text{ ms}$$

$$T_{PF} = 28 \text{ ms} = 28.000.000 \text{ ns}$$

$$\%_{DIRTRBIT_UNO} = 40\% = 0,4$$

$$1 - P_{RF} = 18\% = 0,18 \Rightarrow P_{RF} = 0,82$$

$$T_{SWAP} = 20 \text{ ms} = 20.000.000 \text{ ns}$$

$$P_{R2} = 68\% = 0,68$$

$$\begin{aligned} EAT = & P_{R2} (T_{TLB} + T_{MEM}) + (P_{R1} - P_{R2}) (T_{TLB} + 2T_{MEM}) + \\ & + (1 - P_{R2}) (T_{TLB} + T_{MEM} + T_{PF} + T_{TLB} + 2T_{MEM}) \end{aligned}$$

$$\begin{aligned} EAT = & 0,68 (30 + 350) + (0,82 - 0,68) (30 + 2 \cdot 350) + \\ & + 0,18 (30 + 350 + 28.000.000 + 30 + 2 \cdot 350) \Rightarrow \end{aligned}$$

$$\Rightarrow 258,4 + 102,2 + 504.0199,8 = 504.0560,4 \text{ ms}$$

b) Si supponga che, quando la ricerca nel TLB ha successo, la pagina riferita si trovi nella RAM con il TLB che ha ancora un miss ratio del 13%. Il 9% degli indirizzi logici generati provoca un page fault nel 60% dei casi di page fault la pagina vittima ha il dirty bit a 1. Il tempo medio necessario per trasferire una pagina dalla RAM all'area di swap è di 20 ms.

$$\begin{aligned} EAT = & P_{R2} (T_{TLB} + T_{MEM}) + (P_{R1} - P_{R2}) (T_{TLB} + 2T_{MEM}) + \\ & + (1 - P_{R2}) \cdot \%_{DIRTRBIT_ZERO} \cdot (T_{TLB} + T_{MEM} + T_{PF} + T_{TLB} + 2T_{MEM}) + \\ & + (1 - P_{R1}) \cdot \%_{DIRTRBIT_UNO} \cdot (T_{TLB} + T_{MEM} + T_{PF} + T_{SWAP} + T_{TLB} + 2T_{MEM}) \end{aligned}$$

$$\begin{aligned} EAT = & 0,68 (30 + 350) + (0,82 - 0,68) (30 + 2 \cdot 350) + \\ & + 0,18 \cdot 0,4 \cdot (30 + 350 + 28.000.000 + 30 + 2 \cdot 350) + \\ & + 0,18 \cdot 0,6 \cdot (30 + 350 + 28.000.000 + 20.000.000 + 30 + 2 \cdot 350) \end{aligned}$$

$$\Rightarrow 258,4 + 102,2 + 2.016.079,92 + 5.184.119,88 = 7.200.560,4 \text{ ms}$$

In un sistema che usa paginazione, l'accesso al TLB richiede 40 ms, l'accesso alla memoria 230 ms.

EAT E TLB

29/10/2010

a) Quando si verifica un page fault, si passano 26 ms per caricare in memoria la pagina che si sta cercando. Se il page fault rate è del 13%, il TLB rate è il 78%, indicare l'EAT ai dati.

b) Si supponga che, quando la ricerca nel TLB ha successo, la pagina richiesta si trovi in RAM, con il TLB che ha ancora un miss ratio dell'13%. Si sa che il 7% degli indirizzi logici generati provoca un page fault, e nel 70% dei casi di page fault la pagina virtuale ha il dirty bit a 1. Il tempo medio necessario per trasferire una pagina dalla RAM all'area di swap è di 30 ms.

$$T_{TLB} = 40 \text{ ms}$$

$$\% \text{ DIRTY-BIT-ZERO} = 30\% = 0,3$$

$$T_{MEM} = 230 \text{ ms}$$

$$\% \text{ DIRTY-BIT-UNO} = 70\% = 0,7$$

$$T_{PF} = 26 \text{ ms} = 26,000,000 \text{ ns}$$

$$T_{SWAP} = 30 \text{ ms} = 30,000,000 \text{ ns}$$

$$1 - Pr_1 = 13\% = 0,13 \Rightarrow Pr_1 = 0,87$$

$$1 - Pr_2 = 7\% = 0,07 \Rightarrow Pr_2 = 0,93$$

$$Pr_2 = 78\% = 0,78$$

$$1 - Pr_2 = 13\% = 0,13 \Rightarrow Pr_2 = 0,87$$

$$\begin{aligned} \text{c) } EAT &= Pr_2 (T_{TLB} + T_{MEM}) + (Pr_1 - Pr_2) (T_{TLB} + 2T_{MEM}) + \\ &+ (1 - Pr_1) (T_{TLB} + T_{MEM} + T_{PF} + T_{TLB} + 2T_{MEM}) \end{aligned}$$

$$\begin{aligned} EAT &= 0,78 (40 + 230) + (0,87 - 0,78) (40 + 2 \cdot 230) + \\ &+ 0,13 (40 + 230 + 26,000,000 + 40 + 2 \cdot 230) \end{aligned}$$

$$\Rightarrow 210,6 + 45 + 3 \cdot 120 \cdot 100,2 = 3 \cdot 120 \cdot 355,2 \text{ ms} = 3,12 \text{ ms}$$

$$\begin{aligned} \text{d) } EAT &= Pr_2 (T_{TLB} + T_{MEM}) + (Pr_1 - Pr_2) (T_{TLB} + 2T_{MEM}) + \\ &+ (1 - Pr_1) \cdot \% \cdot \text{DIRTY-ZERO} \cdot (T_{TLB} + T_{MEM} + T_{PF} + T_{TLB} + 2T_{MEM}) + \\ &+ (1 - Pr_1) \cdot \% \cdot \text{DIRTY-UNO} \cdot (T_{TLB} + T_{MEM} + T_{PF} + T_{SWAP} + T_{TLB} + 2T_{MEM}) \end{aligned}$$

$$\begin{aligned} EAT &= 0,78 (40 + 230) + (0,93 - 0,78) (40 + 460) + \\ &+ 0,07 \cdot 0,3 \cdot (40 + 230 + 26,000,000 + 40 + 460) + \\ &+ 0,07 \cdot 0,7 \cdot (40 + 230 + 26,000,000 + 30,000,000 + 40 + 460) \end{aligned}$$

$$\Rightarrow 234,9 + 30 + 506 \cdot 016,17 + 2 \cdot 666 \cdot 037,73 = 3 \cdot 150 \cdot 318,8 \text{ ms}$$

L'accesso al TLB richiede 30 ms

L'accesso alla memoria richiede 350 ms

EAT E TLB

04/06/2010

a) Si presuma 28 ms per caricare in memoria la pagina cercata. Page fault rate è del 18%, TLB rate è del 68%. Indicare EAT

b) TLB miss ratio del 13%. Si sa che il 9% degli indirizzi logici provoca un page fault, nel 60% dei casi la pagina vittima ha il dirty bit a 1. Il tempo medio per il trasferimento è 20 ms.

$$T_{TLB} = 30 \text{ ms}$$

$$\% \text{ DIRR BIT_0} = 40\%$$

$$T_{MEM} = 350 \text{ ms}$$

$$\% \text{ DIRR BIT_1} = 60\%$$

$$TPF = 28 \text{ ms} = 28 \cdot 1000 \cdot 000 \text{ ns}$$

$$T_{SWAP} = 20 \text{ ms} = 20 \cdot 1000 \cdot 000 \text{ ns}$$

$$1 - Pr_1 = 18\% = 0,18 \Rightarrow Pr_1 = 0,82$$

$$1 - Pr_2 = 13\% = 0,13 \Rightarrow Pr_2 = 0,87$$

$$Pr_2 = 68\% = 0,68$$

$$1 - Pr_1 = 9\% = 0,09 \Rightarrow Pr_1 = 0,91$$

$$\begin{aligned} a) EAT &= Pr_2 (T_{TLB} + T_{MEM}) + (Pr_2 - Pr_1) (T_{TLB} + 2T_{MEM}) + \\ &+ (1 - Pr_1) (T_{TLB} + T_{MEM} + TPF + T_{TLB} + 2T_{MEM}) \end{aligned}$$

$$\begin{aligned} EAT &= 0,68 (30 + 350) + (0,82 - 0,68) (30 + 2 \cdot 350) + \\ &+ 0,18 (30 + 350 + 28 \cdot 1000 \cdot 000 + 30 + 2 \cdot 350) \end{aligned}$$

$$\Rightarrow 258,4 + 102,2 + 5 \cdot 0,68 \cdot 199,8 = 5060 \cdot 560,4 \text{ ms} = 5,06 \text{ ms}$$

$$\begin{aligned} b) EAT &= Pr_2 (T_{TLB} + T_{MEM}) + (Pr_2 - Pr_1) (T_{TLB} + 2T_{MEM}) + \\ &+ (1 - Pr_1) \cdot \% \text{ DIRR_0} \cdot (T_{TLB} + T_{MEM} + TPF + T_{TLB} + 2T_{MEM}) \\ &+ (1 - Pr_1) \cdot \% \text{ DIRR_1} \cdot (T_{TLB} + T_{MEM} + TPF + T_{SWAP} + T_{TLB} + 2T_{MEM}) \end{aligned}$$

$$\begin{aligned} EAT &= 0,87 (30 + 350) + (0,91 - 0,87) (30 + 2 \cdot 350) + \\ &+ 0,09 \cdot 0,40 \cdot (30 + 350 + 28 \cdot 1000 \cdot 000 + 30 + 700) + \\ &+ 0,09 \cdot 0,60 \cdot (30 + 350 + 28 \cdot 1000 \cdot 000 + 20 \cdot 1000 \cdot 000 + 30 + 700) \end{aligned}$$

$$\Rightarrow 330,6 + 29,2 + 1 \cdot 0,08 \cdot 0,39,96 + 2 \cdot 592 \cdot 0,59,96 = 3600 \cdot 459,7 \text{ ms}$$

Si consideri la gestione della memoria a partizioni variabili con politica WORST FIT

ALLOCAZIONE DELLA MEMORIA

Processo A occupa le partizioni di origine

21300 40500 62000 e lunghezze
10000 7000 2000 (codice, dati, pila).

02/07/2010

WORST FIT

Processo A esegue una fork che genera processo B.

Al momento della chiamata sono libere 3 partizioni, delle cui dimensioni rispettivamente agli indirizzi

77000 90000 400000 e lunghezze
12000 10000 20000

Indicare:

- 1) l'origine e la lunghezza delle partizioni libere dopo l'esecuzione della fork
- 2) i contenuti dei registri base e dei registri limite dei segmenti di A durante la sua esecuzione
- 3) i contenuti dei registri base e dei registri limite dei segmenti di B durante la sua esecuzione

WORST FIT: assegna il luogo più grande al processo.

BEST FIT: assegna al processo il più piccolo luogo capace di contenere

NEXT FIT: si assegna al processo il primo luogo abbastanza grande da contenere a partire dal punto in cui era terminata la ricerca precedente.

FIRST FIT: si assegna al processo il primo luogo abbastanza grande da contenere a partire dall'inizio della lista.

- 1) PARTIZIONI LIBERE: 89000 → 89000 (5000)
92000 → 100000 (8000)
610000 → 620000 (10000)

2) REGISTRI BASE A : 21300 40500 62000
REGISTRI LIMITE A : 10000 7000 2000

3) REGISTRI BASE B : 77000 90000 400000
REGISTRI LIMITE B : 7000 2000 10000

X

LA gestione della memoria a partizioni variabili con politica FIRST FIT

ALLOCAZIONE DELLA MEMORIA

Processo A occupa le posizioni di origine

31/01/2011

20 000 40 000 62 000 e lunghezze

FIRST FIT

9 000 8 000 5 000 (codice, dati, pila)

Processo A esegue fork che genera processo B
Tre partizioni libere, che indicano agli indirizzi

80 000 90 000 100 000 e lunghezze

8 000 10 000 15 000

I valori sono:

1) origine e lunghezza delle partizioni libere dopo la fork

2) registri base e limite di A

3) registri base e limite di B

WORST FIT: assegna l'area di memoria più grande

BEST FIT: assegna l'area di memoria più piccola capace di contenere il processo

NEAREST FIT: assegna la vicina area di memoria abbastanza grande da contenere il processo a partire dal punto in cui era terminata la ricerca precedente

FIRST FIT: assegna la vicina area di memoria abbastanza grande da contenere il processo a partire dall'inizio della lista.

1) PARTIZIONI LIBERE : 99 000 → 100 000 (1 000)
105 000 → 115 000 (10 000)

2) REGISTRI BASE A : 20 000 40 000 62 000
REGISTRI LIMITE A : 9 000 8 000 5 000

3) REGISTRI BASE B : 80 000 90 000 100 000
REGISTRI LIMITE B : 8 000 9 000 5 000

BLOCCHI FISICI ALLOCATI OGNI 2 PER I
FILE A E B

SCHEDULING DISCO

16/03/2011

PARTE 1

NUMERO TRACCE 400

NUMERO BLOCCHI X TRACCIA 260

DIMENSIONE BLOCCO 4 KB = 4096 Byte

DIMENSIONE INDIRIZZI 32 BIT = 4 Byte

CALCOLO DEL NUMERO DI INDIRIZZI PER BLOCCHI INDIRETTI

BLOCCO INDIRETTO DI PRIMO LIVELLO:

$$4096/4 = 1024 \text{ indirizzi} = 2^{10} \text{ indirizzi}$$

$$\text{BLOCCO INDIRETTO DI } 2^{\circ} \text{ LIVELLO } 2^{20} \text{ indirizzi}$$

$$\text{BLOCCO INDIRETTO DI } 3^{\circ} \text{ LIVELLO } 2^{30} \text{ indirizzi}$$

INODE DEI FILE A E B

INODE FILE A

1	0
2	2
3	4
4	6
5	8
6	10
7	12
8	14
9	16
10	18
11	20
12	22
13	2^{10} indirizzi : 26 -> 2070
14	2^{20} indirizzi : 2072 -> ---
15	2^{30} indirizzi : --- -> ---

INODE FILE B

1	1
2	3
3	5
4	7
5	9
6	11
7	13
8	15
9	17
10	19
11	21
12	23
13	2^{10} indirizzi : 25 -> 2071
14	2^{20} indirizzi : 2073 -> ---
15	2^{30} indirizzi : --- -> ---

TRADUZIONE BLOCCHI LOGICI \rightarrow FISICI

$$(5, A) = 10 - 2 = 8$$

$$(530, B) = 1060 - 1 = 1059$$

$$(10, A) = 20 - 2 = 18$$

$$(9340, B) = 18680 - 1 = 18679$$

$$(15, A) = 30 - 2 = 28$$

$$(4540, B) = 9080 - 1 = 9079$$

$$(2065, A) = 4130 - 2 = 4128$$

$$(6320, B) = 12640 - 1 = 12639$$

CALCOLO DELLE TRACCE

$$8 / 260 = 0$$

$$18 / 260 = 0$$

$$28 / 260 = 0$$

$$4128 / 260 = 15$$

$$1059 / 260 = 4$$

$$18679 / 260 = 71$$

$$90F9 / 260 = 34$$

$$12639 / 260 = 48$$

SCHEDULING
DEL DISCO

16/3/2011

PARTE 2

CALCOLO TEMPO TRASFERIMENTO DI UN BLOCCO

$$\bar{T} = 10 \text{ MB/s}$$

$$10 \text{ MB/s} \cdot 1s = 4 \text{ KB} : x \Rightarrow 10240 \text{ KB} : 1s = 4 \text{ KB} : x \Rightarrow$$

$$10240 \text{ KB} : 1000 \mu\text{s} = 4 \text{ KB} : x \Rightarrow x = \frac{1000 \cdot 4}{10240} = 0,39 \mu\text{s}$$

CALCOLO LATENZA ROTAZIONALE MASSIMA

$$\bar{T} = \frac{1}{2} \cdot \frac{8}{N}$$

$$0,39 = \frac{1}{2} \cdot \frac{1}{260} \Rightarrow \frac{1}{2} = 0,39 \cdot 260 = 101,4 \mu\text{s}$$

OPERAZIONI SUL BLOCCO AGGIUNTIVO (1070, B)

TRAD. IND. LOGICO \rightarrow IND. FISICO : 2140 - 2 = 2139

CALCOLO TRACCIA $\rightarrow 2139 / 260 = 8$

RIEPILOGO

~~BLOCCO~~ DATI

(5, A) (10, A) (15, A) (2065, A) (9340, B) (4540, B) (6320, B)

8 18 28 1059 4128 18679 90F9 12639

0 0 0 4 15 71 34 48

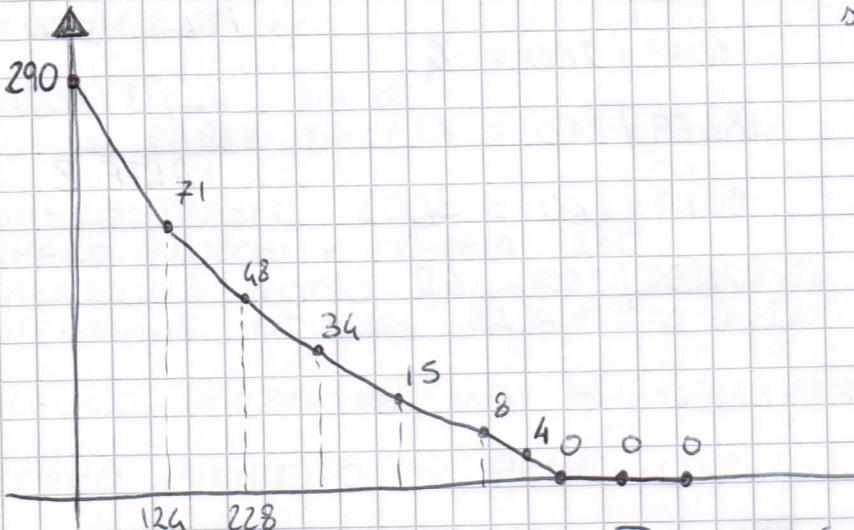
SEEK = 0,1

LATENZA = 101,4 μs

T_{TRASFERIMENTO} = 0,39 μs \approx 0,4 μs

SSTF

Seleziona la Traccia più vicina a quella attualmente schedolata



$$T_A = T_S + T_R + T_T$$

$T_s = m \cdot \text{SEEK}$
 $T_r = \text{lataenza}$
 $T_f = \text{trasferimento}$

$m =$ numero Tracce attraversate

$$T_A [290 - 0.71] = 21,9 + 101,4 + 0,39 = 123,7 \text{ ms}$$

$$T_0 [F1 - D48] = 2,3 + 101,4 + 0,39 = 104,1 + 123,7 = 227,8$$

$$T_A [48 - 0,34] = 1,4 + 101,4 + 0,4 = 102,8 + \underline{227,8} = \underline{\underline{330,8}}$$

$$T_4 [34 - 0,15] = 1,9 + 101,4 + 0,4 = 103,7 + 330,6 = 434,3$$

$$T_A [15-08] = 0,7 + 101,4 + 0,4 = 102,5 + 434,3 = 536,8$$

$$T_A [8 - 0,4] = 0,8 + (0,1 \cdot 4) + 0,4 = 102,8 + 536,8 = 639$$

$$T_A [4 \rightarrow 0] = 102,8 + 639,7 = 225,5 + 639,7 = 741,6$$

$$T_A [0 \rightarrow 0] = 101,8 + 743,6 = 843,4$$

$$T_A[0 \rightarrow 0] = 101,8 + 843,4 = 945,2$$

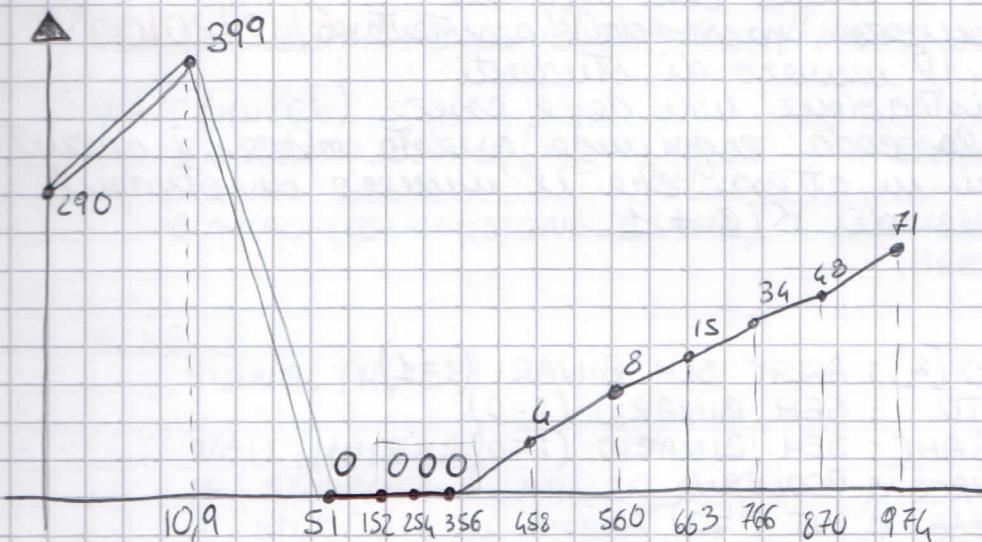


E-SEAN

SCHEDULING DISCO
16/3/2011 PARTE 4

Macchina da scrivere:
Schedula Tutte le Tracce
lungo una direzione

ULTIMO MOVIMENTO 230 -> 290



$$T_A[290 \rightarrow 399] = 109 \cdot 0,1 = 10,9$$

$$T_A[399 \rightarrow 0] = 399 \cdot 0,1 = 39,9 + 10,9 = 50,8 \text{ ms}$$

$$T_A[0 \rightarrow 4] = 101,4 + 0,1 = 101,8 + 50,8 = 152,6 \text{ ms}$$

$$T_A[0 \rightarrow 8] = 101,4 + 0,4 = 101,8 + 152,6 = 254,4 \text{ ms}$$

$$T_A[0 \rightarrow 15] = 101,4 + 0,4 = 101,8 + 254,4 = 356,2 \text{ ms}$$

$$T_A[0 \rightarrow 34] = 0,4 + 101,4 + 0,4 = 102,2 + 356,2 = 458,4 \text{ ms}$$

$$T_A[4 \rightarrow 8] = 0,4 + 101,4 + 0,4 = 102,2 + 458,4 = 560,6 \text{ ms}$$

$$T_A[8 \rightarrow 15] = 0,7 + 101,4 + 0,4 = 102,5 + 560,6 = 663,1 \text{ ms}$$

$$T_A[15 \rightarrow 34] = 1,9 + 101,4 + 0,4 = 103,7 + 663,1 = 766,8 \text{ ms}$$

$$T_A[34 \rightarrow 48] = 1,4 + 101,4 + 0,4 = 103,2 + 766,8 = 870$$

$$T_A[48 \rightarrow 71] = 2,3 + 101,4 + 0,4 = 104,1 + 870 = 974,1$$

N studenti devono sostenere un esame.
Ci sono due docenti e un assistente.

SINCRONIZZAZIONE ESAME UNIVERSIT.

Ogni studente sostiene l'esame con uno dei due docenti e, se lo supera, passa dall'assistente per la registrazione. Il numero di studenti in attesa per la registrazione non deve essere maggiore di K. Se la coda raggiunge questo valore, i docenti sospendono gli esami in attesa che il numero di studenti in attesa sia minore di $K/2 + 1$.

VAR:

DOCENTE_LIBERO[2] : ARRAY SEM BINARI ($:= 1$)
STUDENTE_PRONTO : SEM BINARIO ($:= 0$)
CONCLUSIONE_ESAME : SEM BINARIO ($:= 0$)
RISULTATO_ESAME : BOOLEAN
STUDENTI_IN_ATTESA : INT ($:= 0$)
K : INT
STUDENTE_VOTO : SEM_BINARIO ($:= 0$)
CONFERMA : SEM_BINARIO ($:= 0$)
SITUAZIONE_K : BOOLEAN ($:= FALSE$)
PERMESSO_ASSISTENTE : SEM_BINARIO ($:= 0$)
MUTEX, MUTEX2, MUTEX3 : SEM_BINARI ($:= 0$)

3 processi : STUDENTE, PROFESSORE, ASSISTENTE

STUDENTE()

BEGIN

WAIT (DOCENTE_LIBERO[i])
SIGNAL (STUDENTE_PRONTO)
WAIT (CONCLUSIONE_ESAME) // ATTENDE UN DOCENTE LIBERO
// COMUNICA DISPONIBILITA' AL PROF
// ATTENDE CHE L'ESAME SI CONCLUDA

WAIT (MUTEX)
IF RISULTATO_ESAME == FALSE // ESAME NON SUPERATO
 SIGNAL (MUTEX)
 EXIT() // VAI VIA

ELSE // ESAME SUPERATO
 SIGNAL (MUTEX)
 SIGNAL (STUDENTE_VOTO) // COMUNICA DISPONIBILITA' PER LA
// REGISTRAZIONE ALL'ASSISTENTE

WAIT (CONFERMA) // ATTENDE CHE LA REGISTRAT. SIA EFFETTUATA
EXIT() // VA VIA

END

IONE DOCENTE (i)

BEGIN

WAIT (STUDENTE_PRONTO) // ATTENDE L'ARRIVO DI UNO STUDENTE

RISULTATO_ESAME = ESEGUE_ESAME // VIENE ESEGUITO L'ESAME

WAIT (MUTEX)

IF RISULTATO_ESAME [i] == FALSE // ESAME NON SUPERATO

SIGNAL (MUTEX)

SIGNAL (CONCLUSIONE_ESAME)

// COMUNICA RISULTATO E
LIBERA LO STUDENTE

ELSE

SIGNAL (MUTEX)

WAIT (MUTEX2)

IF STUDENTI_IN_ATTESA < K

STUDENTI_IN_ATTESA ++

SIGNAL (MUTEX2)

// CODA DEGLI STUDENTI < K

// INCREMENTA CODA

WAIT (MUTEX3)

SITUAZIONE_K = FALSE

SIGNAL (MUTEX3)

// FLAG FALSO

SIGNAL (CONCLUSIONE_ESAME) // COMUNICA RISULTATO E
LIBERA LO STUDENTE

ELSE

SIGNAL (MUTEX2)

WAIT (MUTEX3)

SITUAZIONE_K = TRUE

SIGNAL (MUTEX3)

// FLAG VERO

WAIT (PERMESSO_ASSISTENTE)

// SOSPENDE GLI ESAMI, SI
METTE IN ATTESA DEL
PERMESSO DELL'ASSISTENTE
PER RIPRENDERLI

WAIT (MUTEX2)

STUDENTI_IN_ATTESA ++

SIGNAL (MUTEX2)

// HA RIAPRESO GLI ESAMI
INCREMENTA LA CODA

SIGNAL (CONCLUSIONE_ESAME) // RISVEGLIA STUDENTE

END

SI TUTTA

ASSISTENTE ()

BEGIN

WAIT (STUDENTE_VOTO) // ATTENDE UNO STUDENTE CHE DEVE REGISTRARE IL VOTO

WAIT (MUTEX3)

IF SITUAZIONE_K == TRUE // IL PROF HA BLOCCATO GLI ESAMI PERCHÉ LA CODA È PIENA

SIGNAL (MUTEX3)

REPEAT // REGISTRA ESAMI FINO A CHE NON VIENE RAGGIUNTO IL VALORE K/2 + 1

 WAIT (MUTEX)

 STUDENTI_IN_ATTESA --

 SIGNAL (MUTEX)

 SIGNAL (CONFERMA) // LIBERA LO STUDENTE

 UNTIL STUDENTI_IN_ATTESA < K/2 + 1

 SIGNAL (PERMESSO_ASSISTENTE) // RAGGIUNTI I K/2 + 1 PERMETTI AL PROF DI RIPRENDERE GLI ESAMI

ELSE

 SIGNAL (MUTEX3)

 WAIT (MUTEX#)

 STUDENTI_IN_ATTESA --

 SIGNAL (MUTEX)

 SIGNAL (CONFERMA)

END

Composizione di un'unità RAID di 32 GB (spazio utilizzabile di memoria fisica) e con blocco (strip) di 1024 KB, con esercitazione:

RAID

06/12/2010

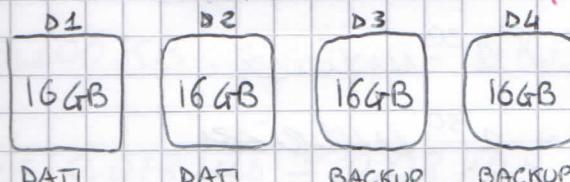
a) configurazione RAID 1 con 4 dischi

b) configurazione RAID 2

Indicare la dimensione effettiva di memoria fisica, in byte, necessaria per la realizzazione.

Considerando un MTTF (disco) = 1500 giorni e MTR (disco) = 5 giorni calcolare il MTTF per le configurazioni A e B

RAPPRESENTAZIONE RAID 1 (a)



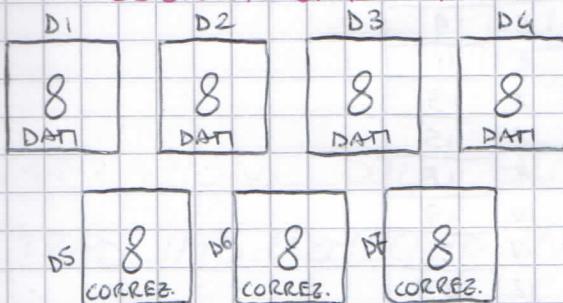
RAID 1 = mirroring

Abbiamo 4 dischi a disposizione, due verranno utilizzati per i dati e due per le copie di backup

$$16 \text{ GB} \times 1024 = 16384 \text{ MB} \times 1024 = 16777216 \text{ KB} \times 1024 = 17179869184 \text{ B}$$

$$17179869184 \times 4 = 68719476736 \text{ B}$$

RAPPRESENTAZIONE RAID 2 (b)



RAID 2 = usa i bit per il CRC in una configurazione che usa 4 dischi dati, occorrono 3 dischi per i bit di correzione

FORMULA GENERALE:

$$\log_2 \text{dischi} + 1 [(\log_2 4) = 3]$$

$$8 \text{ GB} = 8 \cdot 589.936.592 \text{ B}$$

$$[8 \cdot 1024 \cdot 1024 \cdot 1024]$$

$$8 \cdot 589.936.592 \times 7 = 60129562164 \text{ B}$$

N = numero dischi

$$\text{MTTF}_{\text{RAIDO}} = \text{MTTF}_{\text{DISCO}} / N$$

$$\text{MTTF}_{\text{RAID F}} = (\text{MTTF}_{\text{DISCO}})^2 / (2 \times N \times \text{MTR}) = \frac{2 \cdot 250.000}{40} = 56250$$

$$\text{MTTF}_{\text{RAID 2+5}} = (\text{MTTF}_{\text{DISCO}})^2 / (N \times (N-1) \times \text{MTR}) = \frac{2 \cdot 250.000}{210} = 10714$$

BLOCCHI FISICI ALCALATI OGNI 2 PER I
FILE A E B

FILE A PARTE DA 0
FILE B PARTE DA 1

NUMERO TRACCE 400

NUMERO BLOCCHI PER TRACCIA 290

DIMENSIONE BLOCCO 4 KB = 4096 Byte

DIMENSIONE INDIRIZZI 32 bit = 4 Byte

SCHEDULING DISCO

31/1/2011

PARTE 1

CALCOLO DEL NUMERO DI INDIRIZZI PER BLOCCHI INDIRETTI

$$4096/4 = 1024 \text{ indirizzi} = 2^{10} \text{ indirizzi}$$

Blocco indiretto di primo livello = ~~2¹⁰ indirizzi~~ = 2¹⁰ ind.

Blocco indiretto di secondo livello = 2²⁰ indirizzi

"terzo livello" = 2³⁰ indirizzi

INODE FILE A E FILE B

FILE A

1	0
2	2
3	4
4	6
5	8
6	10
7	12
8	14
9	16
10	18
11	20
12	22
13	2^{10} ind : 26 → 2070
14	2^{20} ind : 2072 → --
15	2^{30} ind : --- → --

FILE B

1	1
2	3
3	5
4	7
5	9
6	11
7	13
8	15
9	17
10	19
11	21
12	23
13	2^{10} ind : 25 → 2071
14	2^{20} ind : 2073 → --
15	2^{30} ind : --- → --

TRADUZIONE IND. LOGICI → FISICI

$$(6, B) = (6 \times 2) - 1 = 11$$

$$(4300, B) = (4300 \times 2) - 2 = 8599$$

$$(15, A) = (15 \times 2) - 2 = 28$$

$$(4250, A) = (4250 \times 2) - 2 = 8498$$

$$(34, B) = (34 \times 2) - 1 = 67$$

$$(4560, B) = (4560 \times 2) - 1 = 9119$$

$$(520, A) = (520 \times 2) - 2 = 1038$$

$$(4484, A) = (4484 \times 2) - 2 = 8966$$

CALCOLO DELLE TRACCE

$$22/290 = 0'$$

$$28/290 = 0$$

$$67/290 = 0$$

$$1038/290 = 3$$

$$8599/290 = 29$$

$$8698/290 = 29$$

$$9119/290 = 31$$

$$8966/290 = 30$$

SCHEMING
DISCO

31/1/2011

PARTIE 2

CALCOLO TEMPO DI TRASFERIMENTO DI UN BLOCCO

$$T = 50 \text{ MB/s}$$

$$50 \text{ MB/s} : 1s = 4 \text{ KB} : x \Rightarrow 51200 \text{ KB/s} : 1s = 4 \text{ KB} : x$$

$$\Rightarrow 51200 \text{ KB} : 1000 \text{ ms} = 4 \text{ KB} : x \Rightarrow x = \frac{4 \cdot 1000}{51200} = 0,008 \text{ ms}$$

$$T_f = 0,008 \text{ ms} \approx 0,08 \text{ ms}$$

CALCOLO LATENZA ROTAZIONALE MEDIA

$$T = \frac{1}{2} \cdot \frac{L}{N}$$

L = num. di blocchi da trasferire
 N = num. di blocchi per traccia

$$0,08 = \frac{1}{2} \cdot \frac{L}{290} \Rightarrow \frac{1}{2} = 0,08 \cdot 290 = 23,2 \text{ ms}$$

$$\boxed{\frac{1}{2}} = 23,2 / 2 = 11,6 \text{ ms}$$

OPERAZIONI SUL BLOCCO AGGIUNTIVO

$$\text{TRAD IND LOGICO} \rightarrow \text{IND FISICO} = (1040, A) = (1040 \times 2) - 2 = 2078$$

$$\text{CALCOLO TRACCIA} = 2078 / 290 = 7$$

RIEPILOGO

$$(6, B) (15, A) (34, B) (520, A) (4300, B) (4250, A) (4560, B) (4484, A) (1040, A)$$

$$11 \quad 28 \quad 67 \quad 1038 \quad 8599 \quad 8698 \quad 9119 \quad 8966 \quad 2078$$

$$0 \quad 0 \quad 0 \quad 3 \quad 29 \quad 29 \quad 31 \quad 30 \quad 7$$

$$\text{SEEK} = 0,8$$

$$\text{LATENZA} = 11,6 \text{ ms}$$

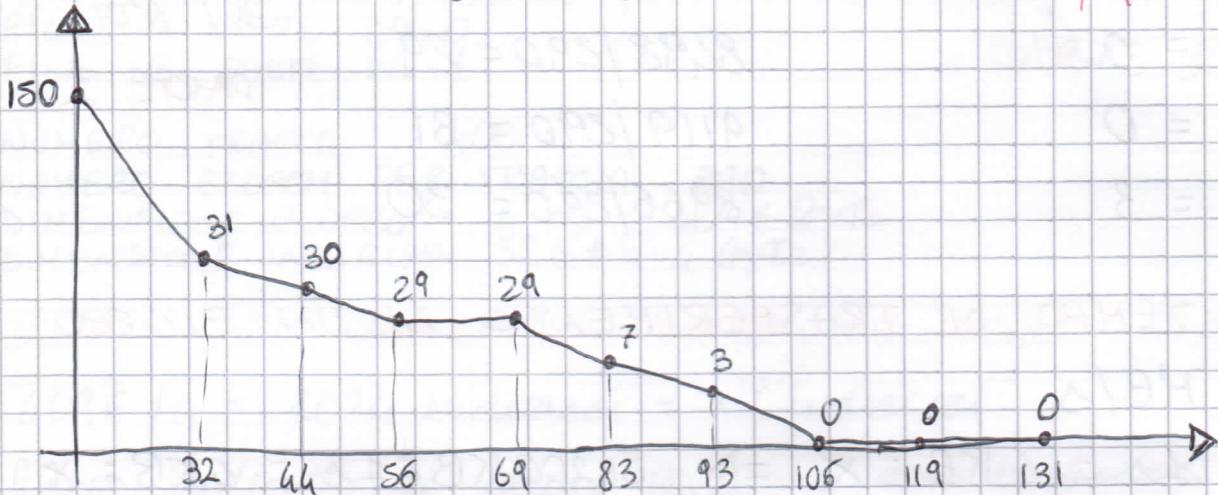
$$\text{TRASFERIMENTO} = 0,08 \text{ ms}$$

SSTF

ULTIMO MOVIMENTO: 120 \rightarrow 150

Seleziona la Traccia più vicina a quella attualmente SCHEDULING DISCO
scheduling

31/1/2011 PARTE 3



$$T_A = T_S + T_R + T_F$$

$$T_S = m \cdot \text{SEEK}$$

$$T_R = \text{Latenza}$$

$$T_F = \text{Trasferimento}$$

m = tracce attraversate

$$T_A[150 \rightarrow 31] = 19,1 + 11,6 + 0,8 = 31,5$$

$$T_A[31 \rightarrow 30] = 0,1 + 11,6 + 0,8 = 12,5 + 31,5 = 44$$

$$T_A[30 \rightarrow 29] = 0,1 + 11,6 + 0,8 = 12,5 + 44 = 56,5$$

$$T_A[29 \rightarrow 29] = 0 + 11,6 + 0,8 = 12,4 + 56,5 = 68,9$$

$$T_A[29 \rightarrow 7] = 2,2 + 11,6 + 0,8 = 14,6 + 68,9 = 83,5$$

$$T_A[7 \rightarrow 3] = 0,4 + 11,6 + 0,8 = 12,8 + 83,5 = 93,6$$

$$T_A[3 \rightarrow 0] = 0,3 + 11,6 + 0,8 = 12,7 + 93,6 = 106,3$$

$$T_A[0 \rightarrow 0] = 0 + 11,6 + 0,8 = 12,4 + 106,3 = 118,7$$

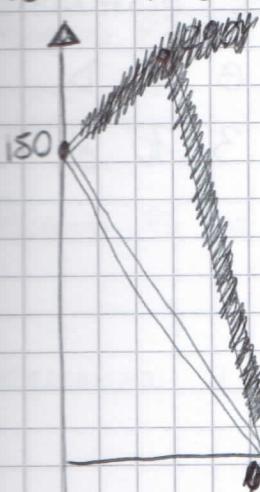
$$T_A[0 \rightarrow 0] = 0 + 11,6 + 0,8 = 12,4 + 118,7 = 131,1$$

ISCO
E 3

C-LOOK

ULTIMO MOVIMENTO:

120 -> 150

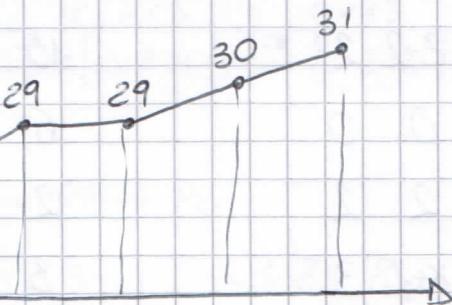


Schedula la Traccia fin
a che ce ve sia da
scheduling lungo la
direzione

SCHEDULING
DISCO 31/11/11

PARTE 4

= spostamento
— scheduling



$$T_A[150 \rightarrow 15] = 15 \text{ ms}$$

$$T_A[0 \rightarrow 0] = 0 + 11,6 + 0,8 = 12,4 + 15 = 27,4$$

$$T_A[0 \rightarrow 0] = 0 + 11,6 + 0,8 = 12,4 + 27,4 = 39,8$$

$$T_A[0 \rightarrow 0] = 0 + 11,6 + 0,8 = 12,4 + 39,8 = 52,2$$

$$T_A[0 \rightarrow 3] = 0,3 + 11,6 + 0,8 = 12,7 + 52,2 = 64,9$$

$$T_A[3 \rightarrow F] = 0,4 + 11,6 + 0,8 = 12,8 + 64,9 = 77,7$$

$$T_A[F \rightarrow 29] = 2,2 + 11,6 + 0,8 = 14,6 + 77,7 = 92,3$$

$$T_A[29 \rightarrow 29] = 0 + 11,6 + 0,8 = 12,4 + 92,3 = 104,7$$

$$T_A[29 \rightarrow 30] = 0,1 + 11,7 + 0,8 = 12,5 + 104,7 = 117,2$$

$$T_A[30 \rightarrow 31] = 0,1 + 11,7 + 0,8 = 12,5 + 117,2 = 129,7$$

Determinare gli intervalli dei valori interi di X, Y e Z per i quali il sistema si trova in uno stato sicuro:

BANCHIERE
STATO SICURO
29/10/2010

RISORSE ALLOCATE

	A	B	C	D
P0	4	Y-1	3	2
P1	8	0	Z-2	2
P2	6	0	0	0
P3	0	1	2	2
P4	2	1	X+1	3

RISORSE MASSIME

	A	B	C	D
	6	4	5	5
	10	1	6	8
	6	2	0	4
	2	3	6	2
	3	2	4	5

RISORSE DISPONIBILI

	A	B	C	D
	2	3	10	8

Determinare gli intervalli dei valori interi di X, Y, Z per i quali il sistema si trova in uno stato sicuro, e se la richiesta di P0 (2, 0, 1, 2) può essere soddisfatta

$$\text{RISORSE NECESSARIE} = \text{RISORSE MASSIME} - \text{RISORSE ALLOCATE}$$

RISORSE NECESSARIE

	A	B	C	D
P0	2	5-Y	2	3
P1	2	1	8-Z	6
P2	2	2	0	4
P3	2	2	4	0
P4	1	1	3-X	2

RIS. DISPONIBILI

	A	B	C	D
	2	3	10	8

Ora occorre trovare un processo per cui $\text{RIS. DISPONIBILI} \geq \text{RIS. NECESSARIE}$

$$\text{Processo P0} \text{ se } \begin{cases} 5-Y \geq 0 \\ 5-Y \leq 3 \end{cases} \Rightarrow \begin{cases} Y \leq 5 \\ Y \geq 2 \end{cases} \Rightarrow 2 \leq Y \leq 5$$

Sommiamo RIS ALLOCATE e RIS DISPONIBILI

$$\text{NUOVE RIS. DISPONIBILI} \quad P0 [6 \quad Y+2 \quad 13 \quad 10] \rightarrow 6 \quad 7 \quad 13 \quad 10$$

$$\text{Processo P1} \text{ se } \begin{cases} 8-Z \geq 0 \\ 8-Z \leq 10 \end{cases} \Rightarrow \begin{cases} Z \leq 8 \\ Z \geq -2 \end{cases} \Rightarrow -2 \leq Z \leq 8$$

$$\text{NUOVE RIS. DISPONIBILI} \quad P1 [14 \quad Y+2 \quad Z+11 \quad 12]$$

Processo P2

nuove ris
disponibili

$$P2 [18 \quad Y+2 \quad Z+11 \quad R2]$$

Processo P3

nuove ris
disponibili

$$P3 [18 \quad X+3 \quad Z+13 \quad 14]$$

Processo P4 se $\begin{cases} 3-x \geq 0 \\ 3-x \leq Z+13 \end{cases} \Rightarrow \begin{cases} x \leq 3 \\ 3x \leq 8+13 \end{cases} \Rightarrow \begin{cases} x \leq 3 \\ x \leq 21-3 \end{cases} \Rightarrow \begin{cases} x \leq 3 \\ x \geq -18 \end{cases} \Rightarrow -18 \leq x \leq 3$

Sequenza sicura

$$P0 \rightarrow P1 \rightarrow P2 \rightarrow P3 \rightarrow P4$$

$$\text{con } -18 \leq x \leq 3 ; \quad 2 \leq Y \leq 5 ; \quad -2 \leq Z \leq 8$$

La richiesta $P0(2,0,1,2)$ può essere soddisfatta immediatamente

Si supponga che in un sistema siano presenti processi P0, P1, P2, P3, P4 ed un insieme di risorse di quattro tipi diversi A, B, C, D e di trovarsi nella seguente configurazione:

BANCHIERE
STATO SICURO

24/10/2007

	RIS ALLOCATE				RIS. MASSIME				RIS DISPONIBILI			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	4	X-1	3	2	6	4	5	6	2	2	10	4
P1	8	0	Y+2	2	10	7	6	8				
P2	4	0	0	0	6	2	0	8				
P3	0	0	3	2	0	3	4	2				
P4	2	1	Z+2	4	9	1	6	9				

Determinare gli intervalli dei valori interi di X, Y, Z per i quali:

- il sistema si trova in uno stato sicuro, eventualmente sequenze secure
- la richiesta di P2 (4,0,0,2) non è stata soddisfatta

$$\text{RIS NECESSARIE} = \text{RIS MASSIME} - \text{RIS ALLOCATE}$$

$$\text{RIS NECESSARIE} \quad \text{RIS DISPONIBILI}$$

	A	B	C	D	A	B	C	D
P0	2	5-X	2	4	2	2	10	4
P1	2	7	8+Y	6				
P2	2	2	0	8				
P3	0	3	1	0				
P4	7	0	5+Z	5				

Ora occorre trovare un processo per cui $\text{RIS DISPONIBILI} \geq \text{RIS NECESSARIE}$

L'unico processo che non è stato soddisfatto è P0 perché:

$$\begin{cases} 5-X \geq 0 \\ 5-X \leq 2 \end{cases} \Rightarrow \begin{cases} X \leq 5 \\ X \geq 3 \end{cases} \Rightarrow 3 \leq X \leq 5$$

Ora si somma le RIS ALLOCATE di P0 a RIS DISPONIBILI

$$\begin{array}{l} \text{NUOVE RISORSE} \\ \text{DISPONIBILI} \end{array} \quad P0 [6 \ x+1 \ 13 \ 6] \rightarrow 6 \ 6 \ 13 \ 6$$

Ora si ripete il procedimento: trovare un processo tale che $\text{RIS DISPONIBILI} \geq \text{RIS NECESSARIE}$ (prendendo in considerazione le nuove risorse disponibili)

L'unico processo è P3. Ora sommiamo RIS ALLOCATE P3 a RIS DISPONIBILI

$$\begin{array}{l} \text{NUOVE RISORSE} \\ \text{DISPONIBILI} \end{array} \quad P3 [6 \ x+1 \ 16 \ 8] \rightarrow 6 \ 6 \ 16 \ 8$$

Ripetiamo il procedimento

Si può soddisfare la richiesta di P2.

Effettuiamo la somma ALLOCATE P2 a DISPONIBILI

$$\begin{array}{l} \text{NUOVE RISORSE} \\ \text{DISPONIBILI} \end{array} \quad P2 [10 \ x+1 \ 16 \ 8] \rightarrow 10 \ 6 \ 16 \ 8$$

Ripetiamo il procedimento

Si può soddisfare P4 se:

$$\begin{cases} 5-z \geq 0 \\ 5-z \leq 16 \end{cases} \Rightarrow \begin{cases} z \leq 5 \\ z \geq 5-16 = -11 \end{cases} \Rightarrow -11 \leq z \leq 5$$

Effettuiamo la somma ALLOCATE P4 + DISPONIBILI

$$\begin{array}{l} \text{NUOVE RISORSE} \\ \text{DISPONIBILI} \end{array} \quad P4 [12 \ x+2 \ z+17 \ 12] \rightarrow 12 \ 7 \ 22 \ 12$$

Ripetiamo il procedimento

Si può soddisfare P1 se:

$$\begin{cases} 8-y \geq 0 \\ 8-y \leq z+17 \end{cases} \Rightarrow \begin{cases} y \leq 8 \\ 8-y \leq z+17 \end{cases} \Rightarrow \begin{cases} y \leq 8 \\ -y \leq z+17-8 \end{cases} \Rightarrow \begin{cases} y \leq 8 \\ y \geq -16 \end{cases} \Rightarrow -16 \leq y \leq 8$$

SEQUENZA SICURA: P0 → P3 → P2 → P4 → P1

$$\text{con: } 3 \leq x \leq 5 \quad -16 \leq y \leq 8 \quad -11 \leq z \leq 5$$

La richiesta di P2 [2, 0, 0, 2] non può essere immediatamente soddisfatta in quanto potrebbe il sistema in uno stato non sicuro. Però essere soddisfatta dopo che P0 e P3 hanno rilasciato le loro risorse.

SCHEDULING

FIFO (senza prelazione) (senza priorità)

I processi sono schedulati nell'ordine in cui giungono al sistema.

SJF - SNPF (senza prelazione) (come senza priorità)

Seleziona il processo in attesa con il CPU-BURST minore. Se due processi hanno lo stesso CPU-BURST viene eseguito quello che è arrivato prima.

In pratica, il CPU-BURST è come se fosse una priorità.

HRRN (senza prelazione)

È come l'SJF-SNPF solo che viene selezionato il processo che presenta il rapporto di risposta maggiore.

$$\text{RAPPORTO DI RISPOSTA} = \frac{W+S}{S}$$

W = è il tempo di attesa
 S = è il CPU-BURST

Il rapporto di risposta va calcolato quando deve essere schedulato un nuovo processo.

SJF - SRTF (con prelazione) (senza priorità)

Funziona come l'SNPF solo che c'è prelazione. In caso di uguaglianza di CPU-BURST viene selezionato il processo che non è stato elaborato per il periodo di tempo più lungo.

RR (con prelazione) (con o senza priorità)

I processi restano in esecuzione per un certo tempo (quanto).

Se non c'è priorità, viene rispettato l'ordine di arrivo.

Se c'è priorità, viene schedulato il processo con priorità maggiore.

N.B. In entrambi i casi, alla fine di ogni quanto, viene effettuato il context switch.

N.B.2 Il quanto non può essere interrotto.

LEN (con relazione)

Seleziona il processo che ha utilizzato il minimo tempo di CPU. In caso di uguaglianza lo scheduler seleziona il processo che non è stato elaborato per il periodo di tempo più lungo.